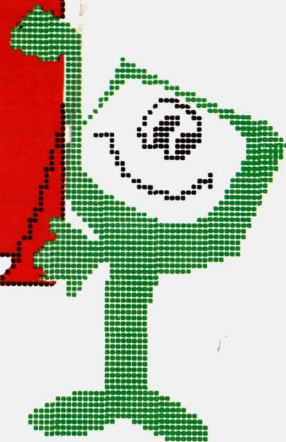
20 VIDEOLEZIONI DI BASIC
PER IMPARARE COL VIC 20

GRUPPO EDITORIALE JACKSON

- La stampante
- Tipi di stampante
- Interfacce per stampanti
- I canali
- OPEN, CLOSE, CMD
- PRINT #
- OUT su stampante
- Videoesercizi
- Videogioco n° 10

10

COMMODORE VIC 20

VIDEO BASIC VIC 20
Pubblicazione quattordicinale
edita dal Gruppo Editoriale Jackson
Direttore Responsabile:
Giampietro Zanga
Direttore e Coordinatore
Editoriale: Roberto Pancaldi
Autore: Softidea - Via Indipendenza 88 - Como
Redazione software:
Francesco Franceschini, Enrico Braglia,
Fabio Calanca
Segretaria di Redazione:
Marta Menegardo
Progetto grafico:
Studio Nuovaidea - Via Longhi 16 - Milano
Impaginazione:
Silvana Corbelli
Illustrazioni:
Cinzia Ferrari, Silvano Scolari
Fotografie:
Marcello Longhini
Distribuzione: SODIP
Via Zuretti, 12 - Milano
Fotocomposizione: Lineacomp S.r.l.
Via Rosellini, 12 - Milano
Stampa: Grafika '78
Via Trieste, 20 - Pioltello (MI)
Direzione e Redazione:
Via Rosellini, 12 - 20124 Milano
Tel. 02/6880951/5

Autorizzazione alla pubblicazione Tribunale di
Milano n° 422 del 22-9-1984
Spedizione in abbonamento postale Gruppo II/70
(autorizzazione della Direzione Provinciale delle
PPTT di Milano).
Prezzo del fascicolo L. 8.000
Abbonamento comprensivo di 5 raccoglitori L. 165.000
I versamenti vanno indirizzati a: Gruppo
Editoriale Jackson S.r.l. - Via Rosellini, 12
20124 Milano, mediante emissione di assegno
bancario o cartolina vaglia oppure
utilizzando il c.c.p. n° 11666203.
I numeri arretrati possono essere
richiesti direttamente all'editore
inviando L. 10.000 cdu. mediante assegno
bancario o vaglia postale o francobolli.
Non vengono effettuate spedizioni contrassegno.

Gruppo Editoriale
Jackson

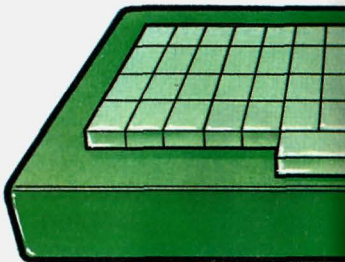
SOMMARIO

Introduzione

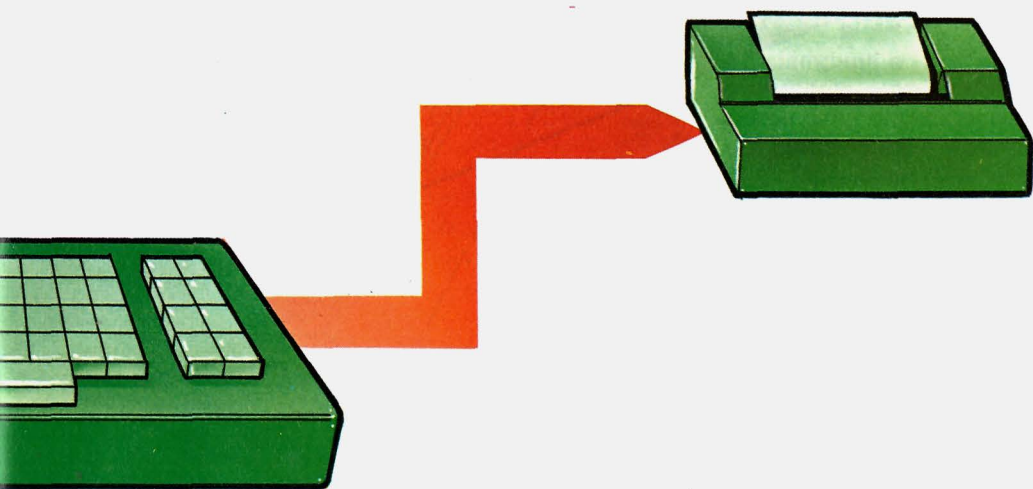
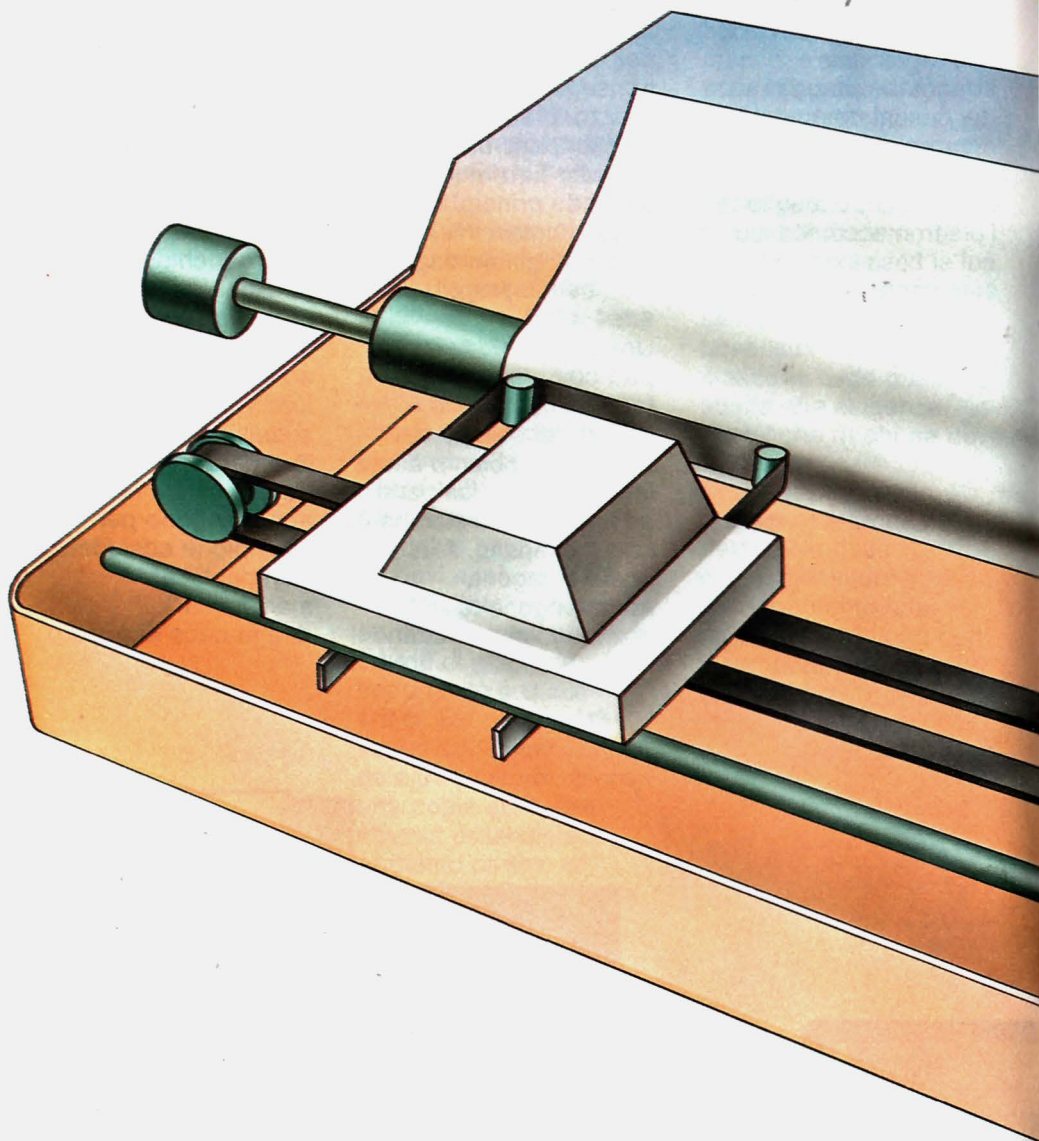
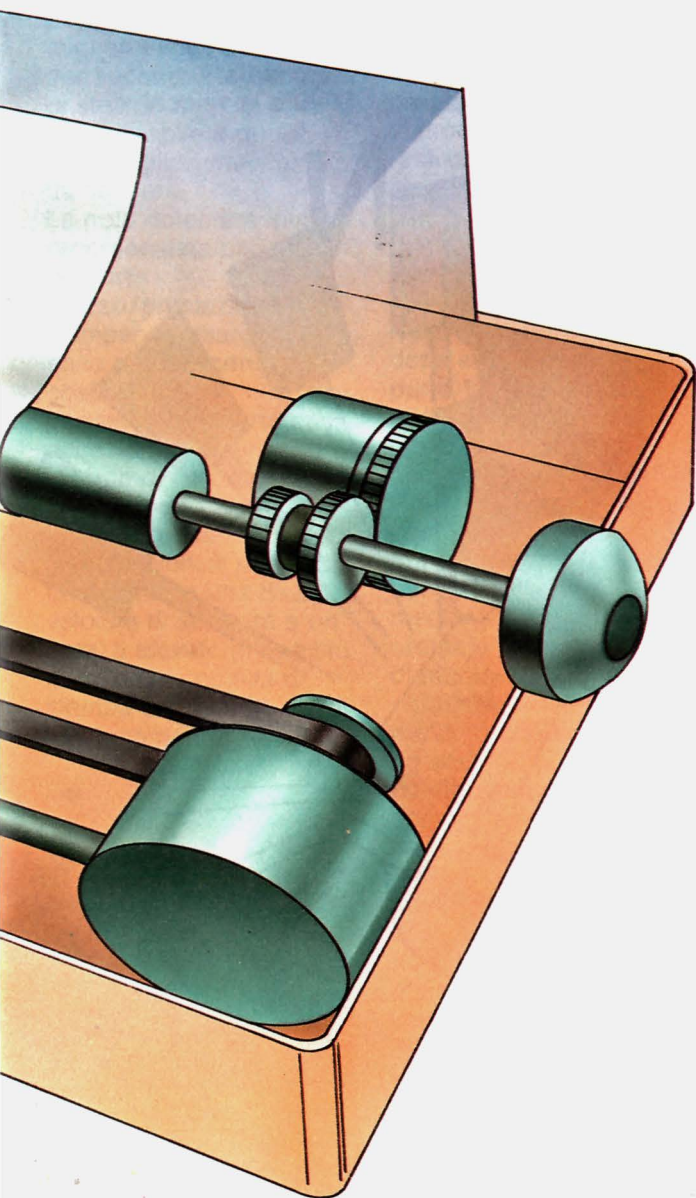
Basta carta, cartaccia, fogli, foglietti; viva la stampante. L'era dei personal computer è stata salutata come la fine del blocco notes: lettere, calcoli, prospetti, direttamente memorizzati su computer, facilmente e rapidamente richiamabili e elaborabili.
E poi? Poi una abituale confortante e pratica copia su carta: la lettera ai clienti, l'elenco dei dischi da fornire a un amico, il programma che, ahimè, non gira.... Si tratta, insomma di eliminare la carta che non serve e di avere invece su carta al momento giusto grazie alla stampante, i dati e le informazioni da analizzare o usare.
Vi sono stampanti per tutti gli usi e per tutte le tasche. Si va dalla "poetica", stampante a margherita, alla versatile stampante ad aghi sino all'economica stampante elettronica.
Ognuna ha caratteristiche proprie di funzionamento, qualità e velocità di stanpa, applicazioni, costo e interfacciamento.
Sono tutti aspetti da conoscere, prima di procedere all'acquisto.

La stampante

La stampante è una delle periferiche più importanti, perché permette di registrare in maniera permanente sulla carta tutti i dati e le informazioni prodotte ed elaborate dal computer.

Grazie alla stampante, infatti, puoi disporre dei risultati delle elaborazioni su un supporto abituale, tangibile, trasportabile e archiviabile; la carta, appunto.
Pensa, ad esempio, alla facilità di dimenticare i dati, una volta letti a video. Se poi i dati sono tanti, non solo il problema aumenta, ma ne insorge un altro: visualizzare i dati tutti assieme, in modo da avere sotto controllo la situazione.
A questo aggiungi che se devi analizzare i risultati, puoi annotare su carta le osservazioni, i punti di maggiore importanza e quelli critici. Se infine i dati vanno consultati anche da altre persone, allora una copia stampata diventa essenziale.
Abbiamo parlato di risultati e dati; le stesse considerazioni però valgono anche per i programmi: un listato è comodo da consultare, puoi correggere gli errori, annotare le eventuali modifiche ed infine hai la sicurezza di non perdere il programma. Il supporto magnetico, infatti, può cancellarsi, accidentalmente o per imperizia; avendo il listato del programma, invece, è sempre possibile ribatterlo.
Una copia stampata risulta pertanto, in parecchie circostanze, di estremo aiuto o utilità, se non addirittura indispensabile.
Naturalmente, esistono stampanti di diversi tipi, dimensioni e prezzi, ciascuna delle quali adatta alle esigenze specifiche di chi ne deve fare uso.
Prima di passare in rassegna le tecniche costruttive delle stampanti, il loro funzionamento e le modalità di dialogo con il computer, apriamo un breve inciso.

HARDWARE

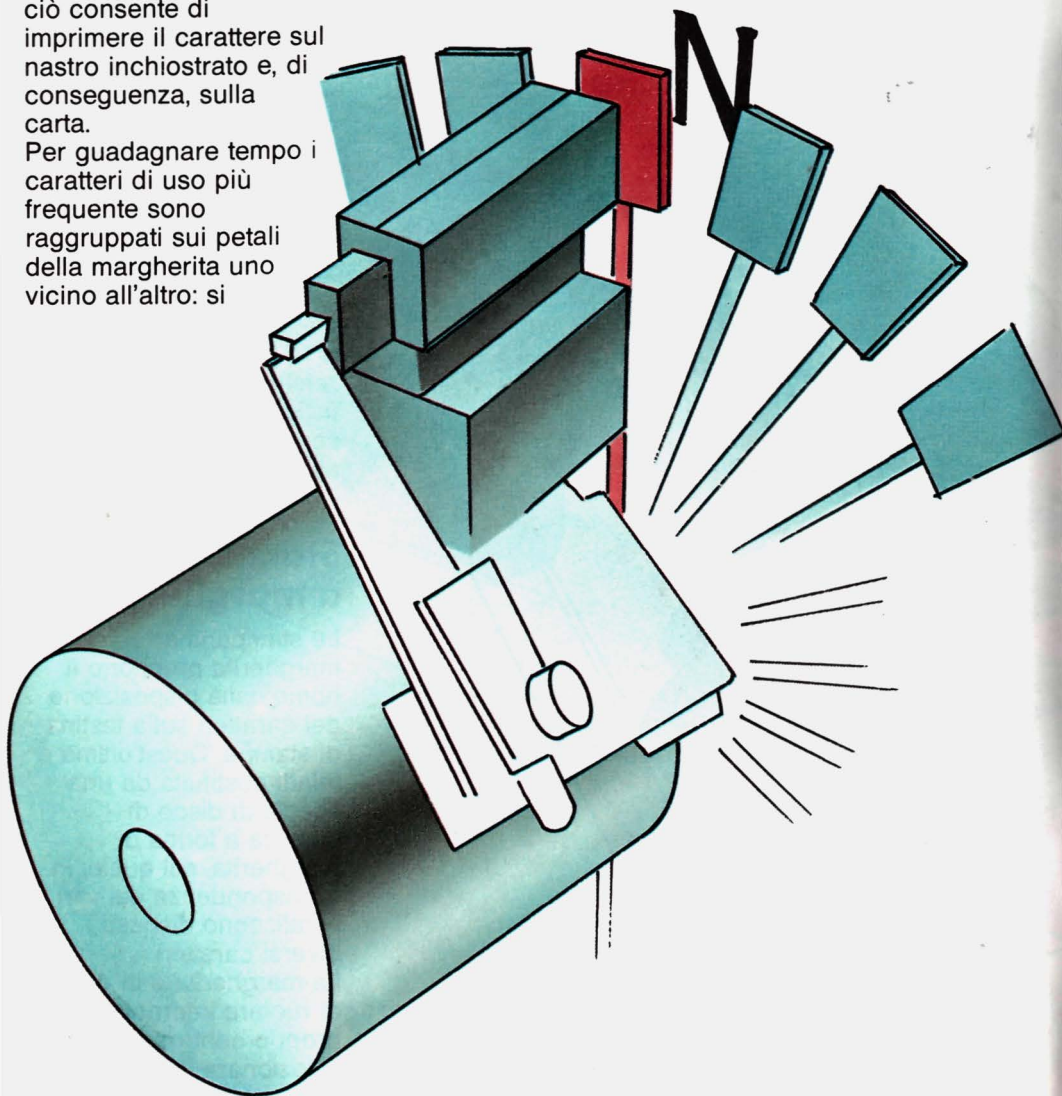
Il prezzo delle stampanti è mediamente elevato, superiore nella stragrande maggioranza dei casi al costo del computer stesso. Il motivo è semplice: la meccanica, o meglio l'elettromeccanica, su cui si basa ogni stampante ha una evoluzione tecnologica molto più lenta dell'elettronica, con conseguente divario prezzo/prestazioni.
Le stampanti esistenti sul mercato funzionano secondo principi fisici spesso notevolmente differenti gli uni dagli altri: ce n'è proprio per tutti i gusti!
Una prima suddivisione può comunque essere fatta considerando il metodo di stampa del carattere: abbiamo allora le cosiddette stampanti ad impatto e le stampanti non ad impatto. Alla prima categoria appartengono le stampanti che sfruttano il più che collaudato metodo della macchina da scrivere: impressionano cioè il carattere sulla carta attraverso la percussione di un martelletto, che alla sommità riporta in rilievo il carattere, sul nastro inchiostrato. In funzione della modalità di formazione del carattere, le stampanti ad impatto possono a loro volta essere suddivise in:

- a carattere intero (a questa categoria appartengono per esempio le cosiddette stampanti "a margherita");
- a matrice di punti o

HARDWARE

"ad aghi".
Le stampanti non ad impatto, invece, per la generazione del carattere fanno riferimento a principi fisici differenti dal nastro inchiostrato: utilizzano una carta sensibile al calore, alla luce o ad un particolare agente chimico/fisico. Tra esse le più diffuse sono:

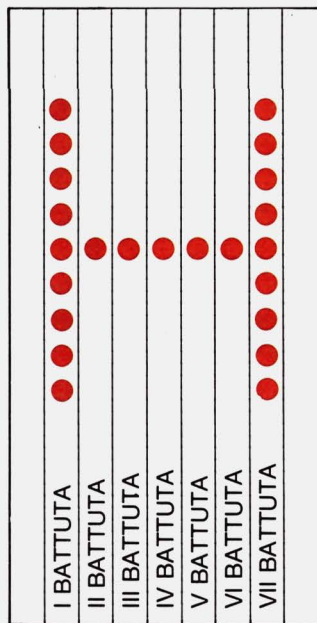
- le stampanti "termiche", che, come indica il nome, sfruttano per la formazione dei caratteri una sorgente di calore;
- le stampanti "elettrostatiche", che utilizzano una carta non sensibile al calore, ma alle scariche elettriche.

Stampanti a margherita

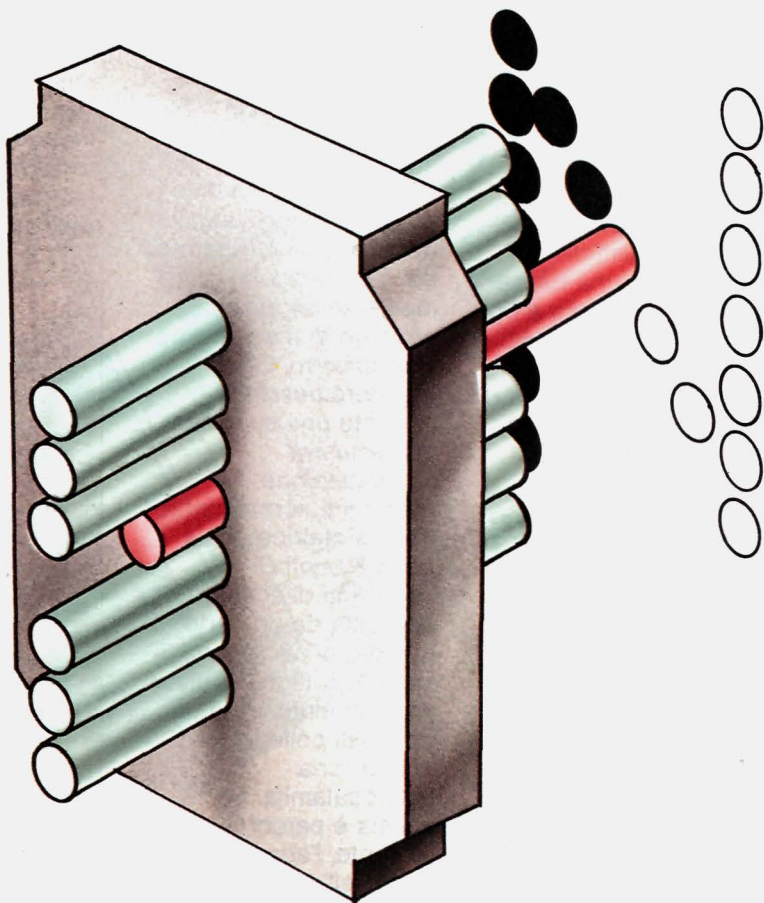
Le stampanti a margherita prendono il nome dalla disposizione dei caratteri sulla testina di stampa. Quest'ultima è infatti costituita da una specie di disco di plastica a forma di margherita, sul quale, in corrispondenza dei vari petali, sono disposti i diversi caratteri.
La margherita è in grado di ruotare intorno al proprio centro per selezionare il carattere

prescelto; quando questo si trova davanti al punto di scrittura viene colpito da un martelletto posto sul retro della margherita ed azionato da una elettrocalamita: ciò consente di imprimere il carattere sul nastro inchiostrato e, di conseguenza, sulla carta.
Per guadagnare tempo i caratteri di uso più frequente sono raggruppati sui petali della margherita uno vicino all'altro: si ottimizzano così le inevitabili pause, dovute alla rotazione della margherita, tra la battitura di due caratteri successivi. Lo stile di scrittura e le dimensioni dei caratteri, inoltre, possono essere modificati con facilità, semplicemente sostituendo la margherita.
Mentre la velocità di

stampa conseguibile dalle macchine di questo tipo non è molto elevata (raggiunge nei casi migliori i 60/70 caratteri per secondo), la qualità di stampa è paragonabile a quella delle migliori macchine da scrivere.
La nota dolente è invece rappresentata dal rapporto prezzo/prestazioni: le stampanti a margherita sono notevolmente costose e non permettono, a parte casi eccezionali, alcuna stampa di tipo grafico.
Morale: sono macchine ideali per aziende, uffici o professionisti disposti ad accettare una bassa velocità di scrittura e un prezzo elevato in cambio di un'ottima qualità di stampa e di affidabilità di funzionamento.

Stampanti ad aghi

Anche questo tipo di stampante deve la propria denominazione al modo in cui viene realizzata la stampa dei singoli caratteri, cioè al processo attraverso il quale si arriva alla formazione dei caratteri sulla carta.
Nella stampante ad aghi, detta anche a matrice di punti, la stampa viene ottenuta in maniera molto simile a quanto accade sullo schermo video: utilizzando una costruzione per punti, si riproduce cioè la forma dei vari caratteri mediante un insieme ordinato di macchioline di inchiostro. Ogni carattere possiede pertanto una specifica ed esclusiva configurazione, composta normalmente da una matrice di 9 x 7 o 6 x 7 punti.
La testina di scrittura è costituita da un certo numero di microscopici martelletti (aghi) allineati verticalmente, ciascuno dei quali collegato ad una propria elettrocalamita. Se questa è percorsa da corrente, l'ago viene spinto all'esterno, fuori dalla punta della testina e, colpendo il nastro inchiostrato, imprime un punto sul foglio di carta.
Le combinazioni di questi punti costituiscono i caratteri.
La testina viene allora fatta scorrere orizzontalmente lungo il foglio e gli aghi stampano ogni carattere.
Supponendo infatti di avere una testina a 9 aghi, con 7 successive battute, ad ogni battuta vengono stampati solo tutti i punti relativi alla linea verticale considerata. Ad esempio, il carattere H viene realizzato così:

La velocità di scrittura, nonostante il maggior numero di operazioni da eseguire, è nettamente superiore a quella della stampante a margherita, arrivando, nei modelli con le migliori prestazioni, ad oltre 250 caratteri per secondo. Di contro, però, peggiora la qualità di stampa, che rimane comunque su livelli decisamente accettabili.

Le stampanti ad aghi sono inoltre molto flessibili, consentendo infatti di scrivere i caratteri in tutte le maniere possibili: largo, stretto, evidenziato, sottolineato, ecc.. Esse presentano oltretutto il non trascurabile vantaggio di poter essere utilizzate come stampanti grafiche.

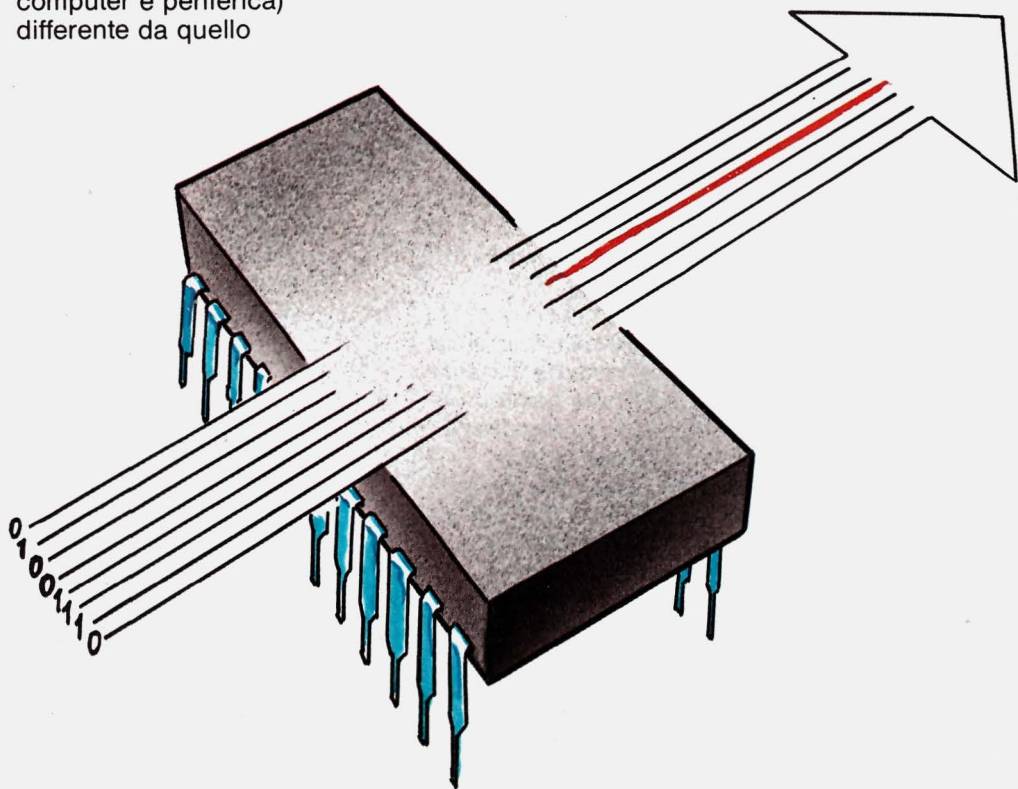
Per quanto riguarda il costo ti basti sapere che l'enorme diffusione che questo tipo di stampante ha avuto, e sta tuttora

avendo, in tutto il mondo dipende in grandissima parte dall'ottimo rapporto tra prestazioni offerte e prezzo richiesto.

Stampanti termiche ed elettrostatiche

Sono le stampanti più economiche e basano il loro funzionamento su meccaniche assai semplici, ma non per questo meno affidabili.
Il principio che porta alla formazione dei caratteri è molto semplice: la carta viene trascinata a velocità costante contro una speciale testina, sulla quale si trova una serie di elementi, che di volta in volta assume la configurazione corrispondente al carattere da stampare.
Tali elementi, estremamente simili a quelli di una stampante ad aghi, quando vengono in contatto con la carta eseguono una particolare azione, che può essere di riscaldamento (nelle stampanti termiche) o di bruciatura (in quelle elettrostatiche).
Naturalmente, la carta su cui avviene la stampa deve poter avvertire questa particolare azione. Occorre perciò adoperare carta appositamente trattata (e quindi più costosa).
La generazione dei caratteri avviene quindi, contrariamente a quanto visto nelle stampanti a margherita e ad aghi, non più a seguito di un "urto" meccanico tra la testina ed il nastro inchiostrato, bensì attraverso altri principi fisici. Da qui la denominazione "non ad impatto".
Anche le stampanti termiche ed elettrostatiche sono estremamente diffuse: nonostante il maggior costo di gestione, determinato dalla carta speciale, esse permettono infatti di ottenere una qualità di stampa più che soddisfacente ad un prezzo contenuto. Inoltre - ed in alcuni casi questo è un vantaggio determinante - lavorano abbastanza silenziosamente.

Interfaccia stampante

La connessione di un'unità periferica al calcolatore non può essere effettuata direttamente: qualunque collegamento deve infatti sempre avvenire attraverso un dispositivo, meglio noto come interfaccia, la cui funzione è quella di fornire tutto il software e l'hardware necessari per le varie operazioni di trasferimento dei dati e delle informazioni.
Anche la stampante non sfugge a questa regola, e richiede quindi che il collegamento si svolga utilizzando un'apposita interfaccia.
La Commodore, per limitare al massimo qualunque disagio e complicazione agli utilizzatori, ha inserito tale interfaccia direttamente all'interno del tuo VIC 20, connettendola all'esterno per mezzo di una porta seriale di ingresso/uscita. Ciò permette anche agli inesperti di collegare l'unità centrale alla stampante senza alcun dispositivo ausiliario e soprattutto senza alcuna difficoltà.

Vi è però il rovescio della medaglia. In un certo senso, la libertà di scelta della stampante è estremamente limitata: in pratica, l'unica (o quasi) stampante compatibile con questa interfaccia è quella originale.
La maggior parte delle stampanti attualmente in commercio, e prodotte da altre case costruttrici, utilizzano infatti un protocollo di comunicazione (cioè uno standard di connessione e "colloquio" tra computer e periferica) differente da quello scelto dalla Commodore per il tuo VIC 20, il che impedisce il collegamento diretto.
Queste stampanti, infatti, sono di solito predisposte per lavorare con uno dei due standard in questo momento più diffusi ed utilizzati: il parallelo (chiamato anche "Centronics") ed il seriale (nella versione RS 232).
Lo standard scelto dalla Commodore per il tuo VIC 20 è invece una semplificazione della versione seriale IEEE 488.
Per poter utilizzare un qualsiasi altro modello di stampante, differente dall'originale, sono quindi necessarie ulteriori modifiche sia al software (con l'inserimento di un programma capace di "comandare" la stampante) che

all'hardware (aggiungendo un'ulteriore interfaccia), allo scopo di adattare il protocollo di comunicazione dell'elaboratore con quello della stampante. La cosa migliore che puoi fare è allora quella di utilizzare la stampante originale, anche in considerazione del fatto che il suo funzionamento - a meno di particolari ed eccezionali esigenze - è più che soddisfacente per tutti gli usi, anche i più impegnativi, ai quali è di norma sottoposta.

Il codice ASCII

Il tuo VIC 20, come d'altronde qualsiasi altro computer, è in grado di comunicare con il mondo esterno soltanto attraverso numeri binari. Tutti i dispositivi periferici, per poter essere connessi ed entrare in contatto con l'unità centrale, devono pertanto adeguarsi (od essere adeguati) a questa caratteristica. Ma c'è di più. Così come l'essere in grado di pronunciare le parole non costituisce condizione sufficiente perché due persone riescano reciprocamente a capirsi (può infatti accadere che ciascuna delle due conosca una lingua diversa da quella parlata dall'altra), anche un calcolatore ed un'unità periferica possono "parlare" - e perciò comprendere - due lingue tra loro più o meno differenti, per

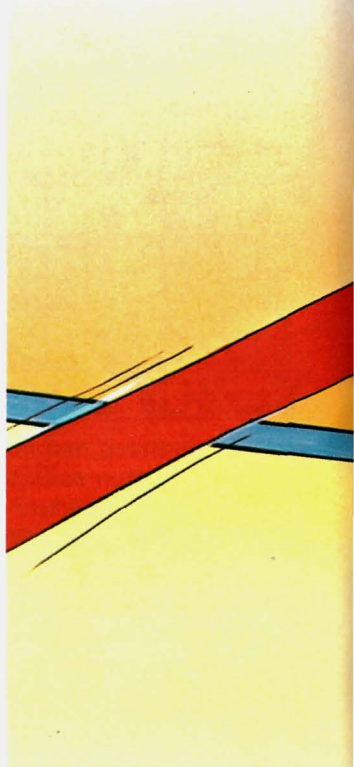
quanto sempre pronunciate in binario, e quindi "non intendersi". Come ben sai, nel campo dei computer anche una leggera differenza nel linguaggio significa di solito una grave possibilità di errore.
È quindi necessario che elaboratore e periferica non soltanto comunichino in binario, ma anche che comprendano e traducano con lo stesso identico significato, senza alcuna possibile differenza, ambiguità od incertezza, le parole usate nella "conversazione".
A questo scopo i costruttori di apparecchiature elettroniche hanno dovuto uniformarsi tra loro, costituendo un vero e proprio standard a cui riferire ciascuna combinazione di numeri binari. Tale standard, identico a quello utilizzato nelle tastiere, si chiama ASCII e permette di assegnare in maniera univoca ad ogni informazione trasmessa un ben determinato codice numerico.
Anche le stampanti ascoltano e parlano il codice ASCII, riuscendo così a comprendere perfettamente tutto ciò che il computer desidera far loro eseguire.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	16 POS	32	48	64	80	96	112	128	144	160	176	192	208	224	240
1	1	17 CRSR DOWN	33	49	65	81	97	113	129	145 CRSR UP	161	177	193	209	225	241
2	2	18 RVS ON	34	50	66	82	98	114	130	146 RVS OFF	162	178	194	210	226	242
3	3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243
4	4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244
5	5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245
6	6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246
7	7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247
8	8 BS	24	40	56	72	88	104	120	136	152	168	184	200	216	232	248
9	9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249
A	10 LF	26 SUB	42	58	74	90	106	122	138	154	170	186	202	218	234	250
B	11	27 ESC	43	59	75	91	107	123	139	155	171	187	203	219	235	251
C	12	28	44	60	76	92	108	124	140	156	172	188	204	220	236	252
D	13 CR	29	45	61	77	93	109	125	141	157	173	189	205	221	237	253
E	14 EN ON	30	46	62	78	94	110	126	142	158	174	190	206	222	238	254
F	15 EN OFF	31	47	63	79	95	111	127	143	159	175	191	207	223	239	255

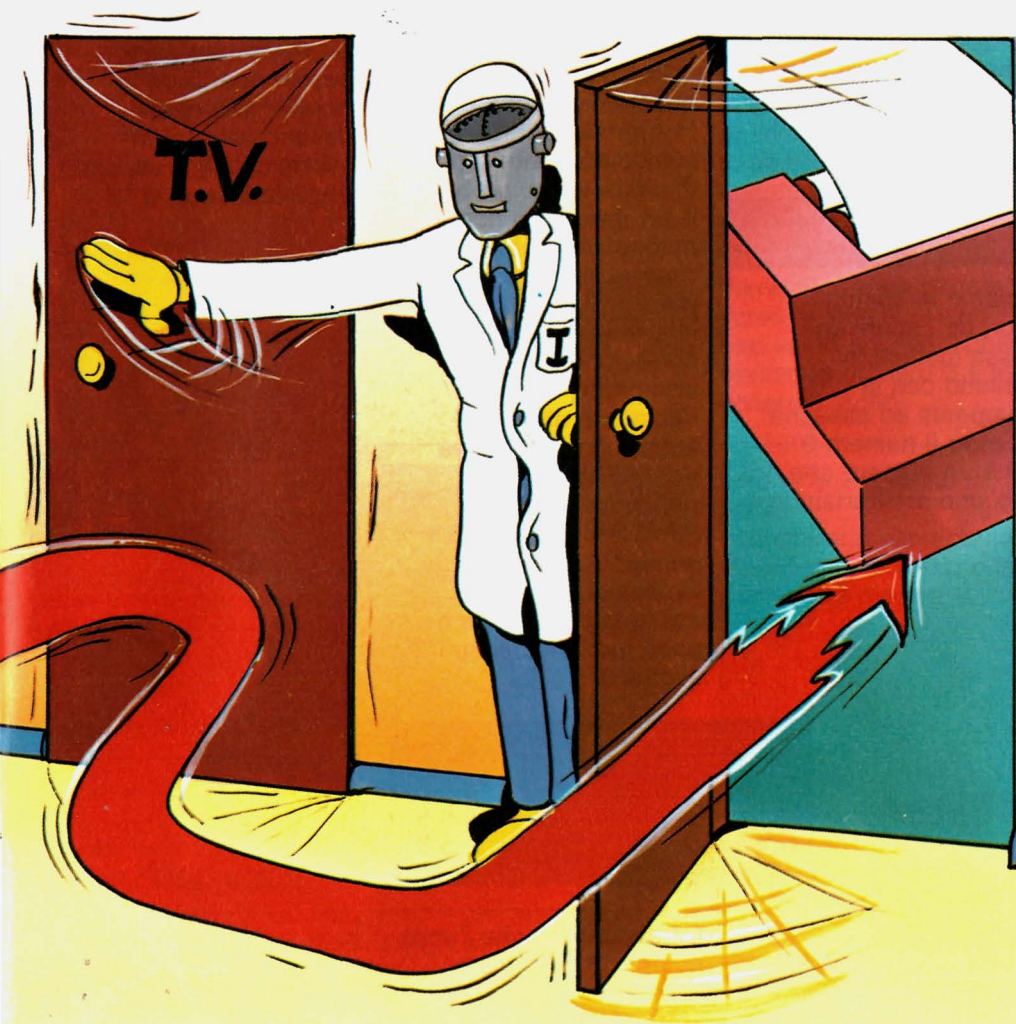
La codificazione dell'intero alfabeto, dei numeri e dei caratteri speciali o di punteggiatura non è comunque sufficiente all'elaboratore per poter avere il completo controllo sulla stampante: occorrono infatti altri codici, che corrispondano, più che a caratteri, ad azioni da eseguire.
Tipici esempi di questi codici di controllo sono i comandi «vai a capo», «salta una riga», «salta una pagina» o «cambia le dimensioni di stampa dei caratteri». Come puoi vedere, non si tratta certo di ordini di secondaria importanza, visto che il loro uso consente, oltre al rispetto delle regole di impaginazione, anche il controllo del movimento meccanico della testina di scrittura o del rullo di trascinamento carta.
Vediamo allora con esattezza i caratteri di controllo disponibili sul tuo VIC 20, per inviare tali ordini alla stampante dedicata.
Avvertenza: se la stampante connessa al tuo computer è diversa da quella originale, può darsi che i codici siano diversi. Sarà allora bene che tu li controlli sul manuale di utilizzo della stampante in tuo possesso prima di adoperarli impropriamente.

CODICE ASCII	EFFETTO STAMPA
8	Permette la stampa di un carattere definito in precedenza nel programma.
10	Il carrello ritorna a capo e salta una riga.
13	Il carrello ritorna a capo.
14	Raddoppia la larghezza dei caratteri (caratteri espansi).
15	Riporta i caratteri alla dimensione normale.
16	Posiziona l'inizio della stampa dei caratteri.
18	Stampa i caratteri in modo reverse (inverso).
145	Passa all'insieme di caratteri grafico/maiuscolo.
146	Annulla la stampa in modo reverse.

I canali

Con il generico termine di "canali" si è soliti indicare i collegamenti che si stabiliscono tra elaboratore e periferica ed attraverso i quali scorrono, proprio come l'acqua in una conduttura, i dati in entrata ed in uscita da una periferica o dall'unità centrale.
Finora ci siamo esclusivamente preoccupati della parte fisica dell'allacciamento tra computer e periferica (nel caso specifico tra computer e stampante), senza avere cioè fatto alcun riferimento all'argomento dal punto di vista software. Adesso è giunto il momento di parlarne.
Computer e stampante, per quanto permanentemente connessi, non sempre sono in reciproca comunicazione. Anzi, per la maggior parte del tempo, visto il relativo e limitato utilizzo della stampante, si disinteressano in maniera assoluta l'uno dell'altra. Quando però arriva il momento occorre che possano capirsi al volo, senza inutili attese o ritardi. Affinché ciò possa avvenire, si è allora pensato di ricorrere ad una specie di rubinetto, da aprire nei momenti più o meno lunghi di comunicazione e da chiudere non appena terminato il flusso dei dati.
Tale funzione di rubinetto è svolta proprio dai canali. Essi vengono allora aperti tutte le volte che un gruppo di dati deve essere inviato o ricevuto (permettendone così il passaggio) e chiusi subito dopo.
Ovviamente, la completa facoltà di decidere se e quando inviare dei dati attraverso un determinato canale occorre sia lasciata all'utilizzatore del computer che dovrà quindi essere messo nella condizione di disporre di specifici comandi, mediante i quali autorizzare od interdire la deviazione delle informazioni verso la stampante (o verso qualche altro dispositivo, visto che il discorso è generale). Spesso capita infatti che la decisione di comunicare con questo o quel dispositivo

dipenda dal verificarsi di certe condizioni: consentire la deviazione e la commutazione dell'invio dei dati è allora, più che un desiderio, una necessità. Vediamo quindi gli strumenti, cioè le istruzioni, che il BASIC ti mette a disposizione per avere pieno ed assoluto controllo sui canali e, di conseguenza, sulla stampante o sulle altre unità periferiche.

OPEN

Questo comando ti permette di aprire un canale di comunicazione, che unisce una generica

periferica con l'unità centrale del tuo VIC 20 ed attraverso il quale i dati possono venire liberamente trasferiti.
OPEN non può essere utilizzato da solo; insieme ad esso è infatti necessario che vengano specificati due numeri: il primo per contraddistinguere il canale ed il secondo la periferica.
Vediamo subito un esempio. Imponendo il comando

```
OPEN 9, 4
```

diciamo al computer: "apri un canale che possa metterti in contatto con la stampante ed assegna ad esso il numero 9".
Il 9 è un numero che noi abbiamo arbitrariamente scelto per contrassegnare quel canale: potevamo infatti utilizzare un qualsiasi altro numero compreso tra 1 e 127. Da adesso in avanti tutte le volte in cui desidereremo compiere qualche operazione che faccia uso del canale appena aperto (cioè della stampante) dovremo sempre e soltanto specificare questo numero: l'interprete BASIC capirà immediatamente e ci ubbidirà senza problemi.
Il 4 è invece un numero particolare, stabilito dalla casa costruttrice: a tale numero il tuo VIC 20 fa infatti automaticamente corrispondere la stampante, senza richiedere ulteriori specifiche, cioè serve per individuare la periferica.
Per tua informazione, i numeri che la Commodore ha assegnato a ciascuna unità periferica sono i seguenti:

Registratore a cassetta	1
Schermo video	3
Stampante	4 o 5
Unità a disco	8 o 9

È bene notare che l'unità numero 3 ha sempre almeno un canale aperto verso l'unità centrale. I dati che vengono inviati allo schermo video percorrono infatti proprio questo canale.
È inoltre importante che tu capisca con esattezza la differenza tra i due numeri subito dopo OPEN: il primo specifica infatti un canale (cioè un percorso che deve essere aperto ed attraverso il quale devono passare le informazioni), il secondo segnala invece la

periferica verso la quale il canale stesso deve dirigersi.
Una periferica può allora connettersi con più canali, mentre un canale non può essere in comunicazione con più di una sola periferica. Se cerchi di aprire un canale già aperto in precedenza, comparirà il messaggio di errore FILE OPEN ERROR.

Esempi

OPEN 1, 4	Apre un canale verso la stampante.
OPEN 2, 4	Apre un secondo canale verso la stampante.
OPEN 13, 4 OPEN 13, 4	La prima OPEN apre un canale verso la stampante. La seconda OPEN provoca invece un messaggio di errore, in quanto si è tentato di aprire due volte (usando lo stesso numero) un canale verso la stampante.
OPEN 8, 4 OPEN 10, 3	Viene aperto un canale verso la stampante ed uno verso il video (l'unità periferica numero 3 è lo schermo video).
OPEN 3, 3	Errore! Si è cercato di aprire contemporaneamente lo stesso canale verso due periferiche diverse.

Sintassi dell'istruzione

OPEN numero1, numero2

PRINT

Una volta aperto, un canale potrà trasferire tutte le informazioni che ad esso faranno riferimento.
Per esempio:

```
OPEN 3, 4
PRINT # 3, "PROVA DI STAMPA"
```

apre un canale (da noi arbitrariamente numerato con 3) verso il dispositivo numero 4, cioè verso la stampante. A tale dispositivo verrà in seguito inviata la stringa di dati "Prova di stampa", visto che l'istruzione PRINT # chiama in causa il canale numero 3.
PRINT # è quindi l'istruzione da impartire per inviare dei dati attraverso uno specifico canale. Devi prestare molta attenzione al fatto che PRINT # e PRINT non sono la stessa cosa, anche se appaiono molto simili e seguono le stesse regole di grammatica BASIC.
PRINT provoca infatti la visualizzazione dei dati sullo schermo video; PRINT # indirizza invece le informazioni verso il canale specificato dal numero ad esso immediatamente seguente. Cercando di inviare informazioni su di un canale non aperto in precedenza, apparirà il messaggio di errore FILE NOT OPEN ERROR.

Esempi

```
OPEN 2, 4
PRINT # 2; A + B
```

Apre un canale verso la stampante e stampa il risultato della somma di A + B.

```
OPEN 1, 4
PRINT "GNAM ..."
PRINT # 1, "GNAM ..."
```

La stringa viene inviata sia al video che alla stampante (canale numero 1).

```
OPEN 3, 4
```

Apre un canale verso la stampante.

```
OPEN 4, 4
```

Apre un secondo canale verso la stampante.

```
OPEN 7, 3
```

Apre un canale verso il video.

```
PRINT "STAMPA SU VIDEO"
PRINT # 3, "STAMPA SU CARTA"
PRINT # 4, "STAMPA SU CARTA"
PRINT # 7, "STAMPA SU VIDEO"
```

Le due stampe sul video e le due stampe sulla carta vengono ottenute utilizzando complessivamente 4 canali: i tre aperti poco prima ed il normale canale di connessione con il video.

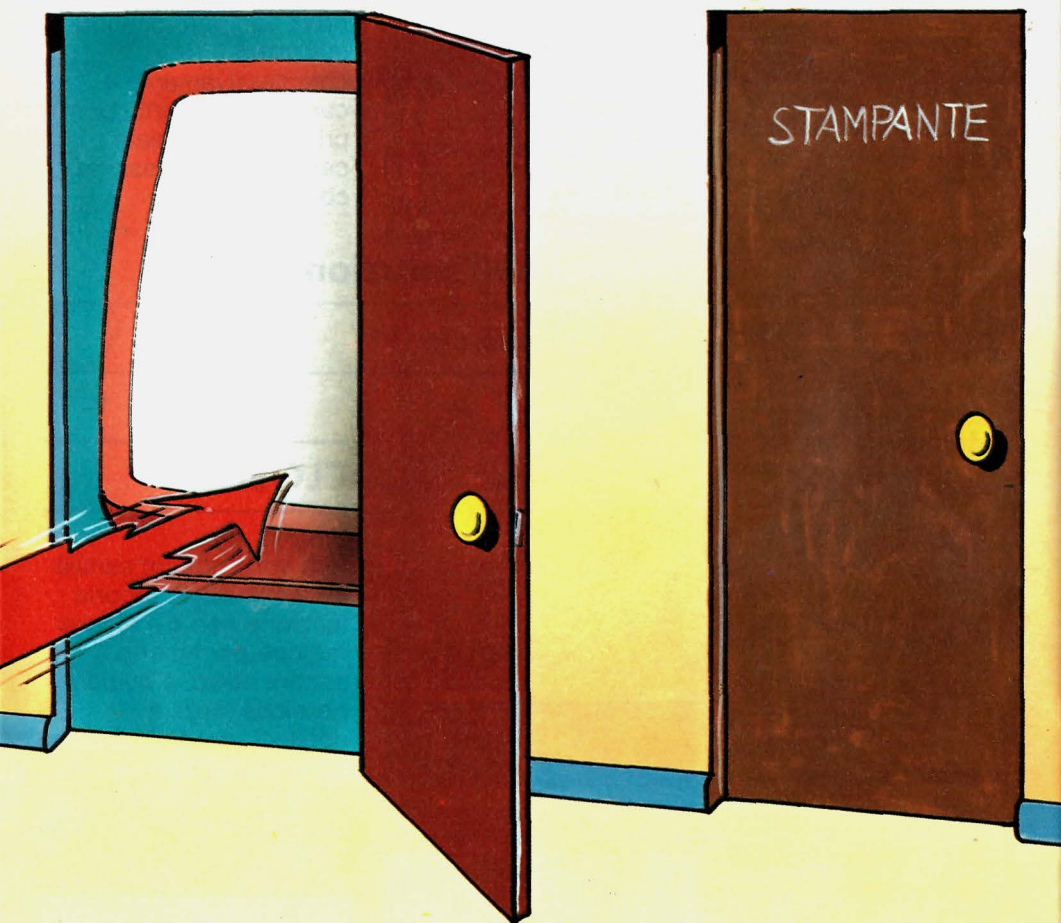
Sintassi dell'istruzione

PRINT # numero, dati

CLOSE

Una volta che un canale ha terminato la propria funzione non c'è alcuna ragione per tenerlo ancora attivo: è quindi possibile (anzi, è obbligatorio) ordinare al computer di scollegarlo attraverso l'istruzione CLOSE.
Così:

```
CLOSE 5
```

dice al tuo VIC 20: "visto che non ne hai più bisogno, chiudi il canale numero 5". Proprio come fai appendendo la cornetta del telefono. Naturalmente, il numero subito dopo la parola CLOSE deve riferirsi ad un canale che in precedenza era già stato aperto.
OPEN e CLOSE viaggiano quindi sempre a braccetto: per ogni OPEN una CLOSE e viceversa.
Qualsiasi programma dovrà allora contenere tante OPEN quante sono le CLOSE:

dimenticandosi di chiudere alcuni canali, il computer li lascerà in sospeso. Il comando RUN chiude automaticamente qualsiasi canale.

Esempi

```
OPEN 3, 4
PRINT # 3, CHR$ (13)
CLOSE 3
```

Apre un canale verso la stampante e fa avanzare di una riga il rullo di trascinamento della carta (13 è il codice ASCII di "ritorno carrello").

```
OPEN 2, 3
PRINT # 2, "VA SULLO SCHERMO"
CLOSE 2
```

Apre un secondo canale (oltre a quello standard) verso lo schermo video (unità periferica numero 3). Il comando di stampa viene quindi inviato attraverso tale canale.

```
OPEN 1, 4
OPEN 2, 3
PRINT "IO VADO SULLO SCHERMO"
PRINT # 1, "IO VADO SULLA STAMPANTE"
PRINT # 2, "ANCH'IO VADO SULLO SCHERMO"
CLOSE 1
CLOSE 2
```

Apre due canali: uno verso la stampante ed uno verso lo schermo video. Il PRINT ha effetto sul video (utilizzando il canale standard); il primo PRINT # va invece alla stampante (sul canale 1); il secondo PRINT # esegue la stampa di nuovo sul video (per quanto utilizzando un canale diverso dalla precedente PRINT).

```
OPEN 11, 4
PRINT # 1, "ADESSO C'È UN ERRORE"
CLOSE 1
```

Mentre le prime due istruzioni sono corrette, la terza provoca la visualizzazione del messaggio di errore FILE NOT OPEN ERROR, avendo cercato di chiudere il canale numero 1 che non era stato aperto. Anziché 1, dopo CLOSE dovrebbe esserci 11.

Sintassi dell'istruzione

CLOSE numero canale

CMD

Supponi di voler scrivere un programma che *debba produrre risultati* solo su stampante. Dovresti allora inserire un'istruzione OPEN che *apra un canale verso la* stampante e tante istruzioni PRINT # quanti saranno i dati da stampare. Esiste però nel BASIC del tuo VIC 20 un comando che *permette di evitare tutta* questa trafila, consentendoti di specificare una volta *sola il canale di* destinazione e di utilizzare da lì in avanti semplici istruzioni PRINT per inviare i dati alla stampante. Tale comando è CMD (abbreviazione dell'inglese Command). Vediamolo subito all'opera:

```
10 OPEN 13, 4: REM APRE UN CANALE
   VERSO LA STAMPANTE
20 CMD 13 : REM DA ADESSO IN AVANTI
   LE PRINT VANNO ALLA STAMPANTE
30 PRINT "PRIMA STRINGA"
40 PRINT "SECONDA STRINGA"
50 PRINT "TERZA STRINGA"
60 PRINT # 13, "ULTIMA STRINGA"
   : REM ULTIMA STAMPA SU CARTA
70 PRINT "QUESTO VA SUL VIDEO"
80 CLOSE 13
90 END
```

CMD lavora in modo che all'esecuzione del comando la linea di comunicazione verso lo *schermo video venga* commutata sul canale numero 13. Da quel momento in poi tutte le PRINT avranno effetto sulla sola stampante.

Il collegamento resta valido fino a quando *(riga 60) viene eseguito* un PRINT # 13.
Questa istruzione riporta infatti al normale *funzionamento di* PRINT, che da lì in avanti tornerà a trasferire le stampe sul video. Il comando CLOSE della linea 80 chiude quindi definitivamente il canale verso la stampante.
L'istruzione PRINT # avvisa la stampante che la comunicazione sta per essere chiusa: questa istruzione deve sempre essere utilizzata quando si usa CMD: prima di chiudere (CLOSE) il canale occorre infatti terminare l'attesa di ulteriori dati in arrivo da parte della periferica.
Dimenticando il PRINT # (cioè passando direttamente a CLOSE) si potrà avere un cattivo funzionamento del video e la stampante non sarà stata "chiusa" correttamente.
Il più frequente utilizzo di *CMD lo si ha per* ottenere i listati dei programmi. Le seguenti istruzioni, impartite sia in *modo diretto che da* programma, ti consentiranno di effettuare proprio questa operazione.

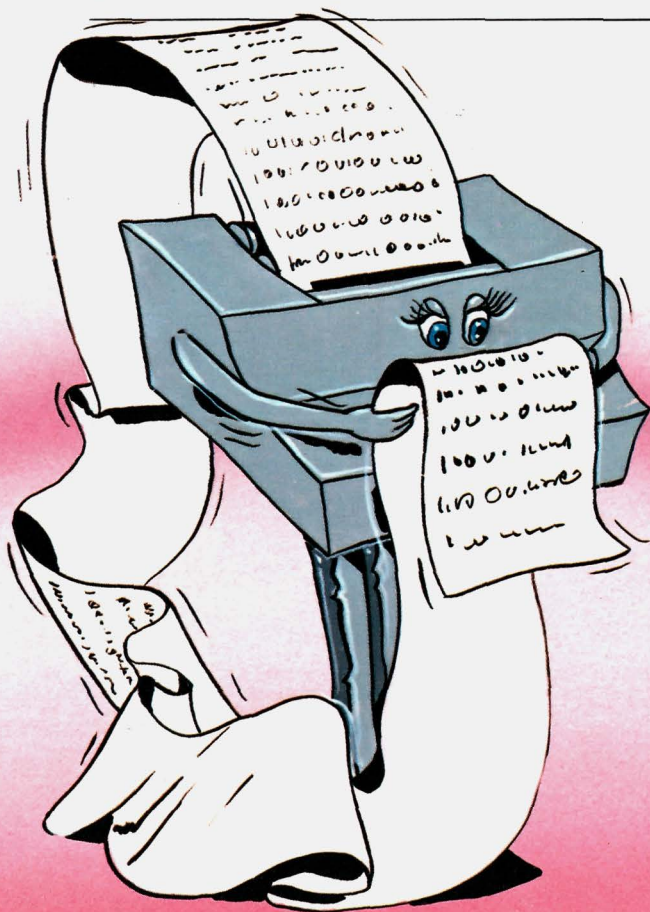
LINGUAGGIO

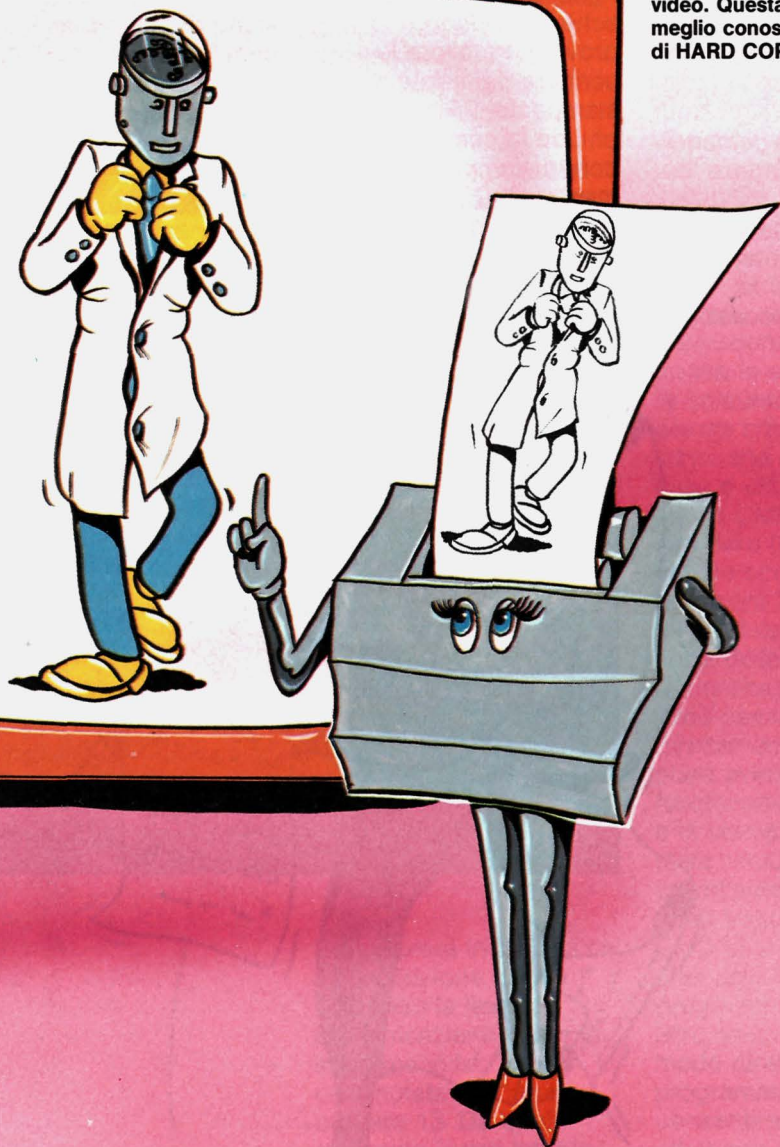
```
OPEN 1, 4
CMD 1
LIST
PRINT # 1
CLOSE 1
```

La sequenza di comandi comincia con l'aprire (OPEN) verso il dispositivo *numero* 4 (cioè la stampante) un canale (1), a cui fa riferimento il successivo CMD. Il LIST, anziché inviare come al solito le *righe di programma* allo schermo video, provocherà allora la scrittura del listato su stampante. PRINT # 1 chiude la comunicazione con la stampante, mentre CLOSE 1 chiude il canale aperto con l'istruzione OPEN.
Se l'istruzione CMD viene riferita ad un canale non aperto, il tuo VIC 20 ti invierà il messaggio:

```
FILE NOT OPEN ERROR
```

Sintassi dell'istruzione

CMD numero

PROGRAMMAZIONE

Con particolari procedure è possibile riprodurre su stampante l'immagine del video. Questa operazione è meglio conosciuta con il nome di HARD COPY.

Out su stampante

Abbiamo detto che la stampante presenta il non trascurabile vantaggio di permettere la scrittura su carta dei risultati delle elaborazioni. Tale vantaggio assume maggior rilievo soprattutto in corrispondenza di quei risultati che non sono immediatamente comprensibili, oppure che possono risultare utili da conservare per un successivo utilizzo.
Il programma che segue sembra fatto apposta per ribadire questo concetto; la sua uscita è infatti rappresentata da una tabella di conversione decimale/binaria.
Ti ricordi i numeri binari? Sono quelle lunghe sequenze di 1 e di 0 che il tuo VIC 20 utilizza per effettuare tutte le operazioni di cui è capace e per ricordare i dati che tu gli suggerisci. Una tabella decimale/binaria è allora proprio ciò che serve: infatti, quasi nessuno è in grado di ricordare a memoria questi numeri, che talvolta risultano utili per impartire all'elaboratore particolari comandi od azioni.
Eccone il listato.

```
10 A$ = CHR$ (13) : REM IN ASCII È UN RITORNO CARRELLO
20 OPEN 1, 4 : REM STAMPANTE
30 PRINT # 1, TAB (26); "CONVERSIONE DECIMALE/BINARIA"
40 PRINT # 1, A$, A$, A$ : REM 3 LINEE BIANCHE
50 PRINT # 1, TAB (10); "DECIMALE"; TAB (40); "BINARIO"
60 PRINT # 1, A$
70 FOR J = 1 TO 255
80 PRINT # 1, TAB (13); STR$ (J);
85 B$ = " " : N = J
90 FOR I = 7 TO 0 STEP - 1
95 B = INT (N/2 ↑ I)
100 B$ = B$ + STR$ (B)
110 N = N - B * 2 ↑ I
120 NEXT I
125 PRINT # 1, TAB (40); B$
130 GET B$
140 IF B$ <> " " THEN 160
150 NEXT J
160 CLOSE 1
170 END
```

Vediamo quindi il funzionamento del programma, peraltro veramente molto semplice.

Le linee 10-60 aprono un canale verso la stampante ed effettuano una sorta di intestazione per il successivo gruppo di risultati: anche l'occhio vuole la sua parte!
Alla linea 70 inizia il ciclo principale: visto che il tuo computer usa numeri binari a 8 cifre, la conversione è stata limitata a 255, che è infatti il massimo valore decimale raggiungibile da un numero binario a 8 bit. Il ciclo verrà quindi ripetuto per 255 volte.
Le linee 90-120 contengono un secondo ciclo - nidificato nel primo - che effettua la *conversione vera e propria*.
Esse operano in questo modo. Supponi che il numero da trasformare sia 170 (cioè J = N = 170). Innanzitutto viene calcolato ed assegnato alla variabile B il numero binario posto nella prima delle 8 posizioni: B assume quindi il valore 1.
Infatti:

INT (N/2 ↑ I) = INT (170/2 ↑ 7) = INT (170/128) = INT (1.32 ...) = 1.

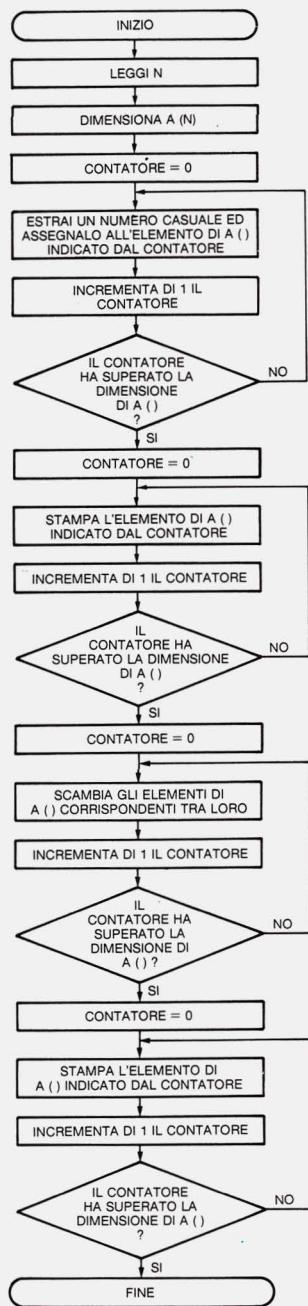
Alla stringa B$ (linea 85), inizialmente vuota, viene perciò assegnato questo valore, debitamente convertito in alfanumerico (linea 100).
Alla linea 110, N, cioè 170, diminuisce di

B * 2 ↑ I = 1 * 2 ↑ 7 = 128

diventando quindi 42.
Nel secondo ciclo I assume valore 6 (STEP − 1) e B valore 0:

INT (N/2 ↑ I) = INT (42/2 ↑ 6) = INT (42/64) = INT (0.65 ...) cioè 0

che la stringa B$ aggiunge all'1 assegnatole in precedenza e così via.
Alla fine del ciclo B$ vale 10101010, che, guarda caso, è proprio il

valore binario di 170.
La funzione di B$ sembrerebbe comunque incomprensibile: perché ricorrere ad essa, visto che di volta in volta abbiamo già il valore della cifra binaria contenuto in B? Presto detto. Se sommassimo man mano i numeri in B, il risultato, essendo la variabile B di tipo numerico, sarebbe ben diverso da quello desiderato: 1 + 1 sommati algebricamente danno 2, mentre sommati come stringhe danno 11.
La linea 125 stampa infine il valore contenuto in B$, affiancandolo al corrispondente valore decimale stampato in precedenza.
Le linee 130-140 sono state inserite per consentirti, nel caso tu lo desiderassi, di interrompere l'esecuzione "via software" senza dover ricorrere a manovre "hardware", come spegnere la stampante.
Schiacciando un qualsiasi tasto provocherai infatti l'immediato arresto del programma. Un buon programma deve sempre prevedere la possibilità di interruzione dell'esecuzione.
Passiamo ora ad un altro problema. Vogliamo provare a "ribaltare" la collocazione degli elementi di un array, mettendo cioè il contenuto della prima posizione nell'ultima e quello dell'ultima nella prima; il secondo nella penultima, il terzo nella terzultima, eccetera.
Desideriamo inoltre che il nostro VIC 20 faccia proprio tutto: creare un array, riempirlo di elementi numerici scelti a caso (viene allora preziosa RND), visualizzarlo in questo ordine, ribaltarlo e visualizzarlo nel nuovo ordine, permettendoci di confrontare i due insiemi di valori. Il programma deve inoltre includere la possibilità di visualizzare i risultati sia su video che su stampante: gli array con molti elementi finirebbero infatti col riempire completamente lo schermo, impedendo qualsiasi controllo.
Vediamo lo schema a blocchi:

ed il corrispondente listato:

```
10 REM INVERSIONE DI UN VETTORE
20 INPUT "QUAL È LA DIMENSIONE DEL VETTORE"; N
30 INPUT "L'USCITA LA VUOI SU VIDEO O STAMPANTE (V/S)"; R$
40 R$ = LEFT$ (R$, 1) : REM PRENDE SOLO IL PRIMO CARATTERE
   DI R$
50 IF R$ < > "V" AND R$ < > "S" AND R$ < > "v" AND R$ < > "s"
   THEN GOTO 30
60 IF R$ = "S" OR R$ = "s" THEN OPEN 1, 4 : REM CANALE
   ALLA STAMPANTE
70 IF R$ = "V" OR R$ = "v" THEN OPEN 1, 3 : REM CANALE
   AL VIDEO
80 DIM A (N) : REM DIMENSIONA L'ARRAY
90 REM CARICA IL VETTORE
100 FOR I = 0 TO N
110 A (I) = INT (RND (0) * N) : REM I NUMERI DELL'ARRAY SARANNO
    COMPRESI TRA 0 ED N
120 NEXT I
130 REM VISUALIZZA IL VETTORE
140 PRINT # 1, "ELEMENTI ORIGINARI"; CHR$ (13);
150 FOR I = 0 TO N
160 PRINT # 1, A (I); "–";
170 NEXT I
180 REM ----------------------------------------
190 REM DA QUI COMINCIA L'INVERSIONE
200 FOR I = 0 TO INT ((N + 1)/2)
210 S = A (I)
220 A (I) = A (N – I)
230 A (N – I) = S
240 NEXT I
250 REM ARRIVATI QUI L'ARRAY È STATO INVERTITO
260 REM ----------------------------------------
270 REM STAMPA L'ARRAY INVERTITO
280 PRINT # 1, CHR$ (13); CHR$ (13); "ELEMENTI INVERTITI"; CHR$ (13)
290 FOR I = 0 TO N
300 PRINT # 1, A (I); "–";
310 NEXT I
320 CLOSE 1
330 END
```

Lo schema a blocchi ed i numerosi commenti rendono inutile ogni ulteriore spiegazione.

Vi sono comunque due cose meritevoli di essere notate:
1) per commutare l'uscita sul video o sulla stampante è sufficiente chiedere in risposta dalla tastiera la opportunità prescelta (vedi le linee 30-60 e 70): a quel punto - aprendo il canale corrispondente alla periferica selezionata - si è allora in grado di soddisfare con estrema facilità ad una qualsiasi delle due possibilità;
2) per scambiare tra loro i due valori contenuti in due variabili (nel nostro esempio tra le coppie di elementi dell'array) è necessario ricorrere ad una terza variabile, il cui incarico è quello di conservare temporaneamente il valore di una delle due variabili di partenza.
Vediamo un esempio:

| 3 | A | 8 | B	Desideriamo scambiare i valori contenuti in queste due variabili
| 3 | A | 8 | B | | C	Prendiamo una terza variabile
| 3 | A | 8 | B | 3 | C	Copiamo nella terza variabile il contenuto di una delle altre due (per esempio della prima)
| 8 | A | 8 | B | 3 | C	Le prime due variabili ora possono diventare uguali: C si conserva inalterata
| 8 | A | 3 | B | 3 | C	Ricopiando C nella seconda variabile, lo scambio è stato eseguito

Le istruzioni che nel nostro programma eseguono questa operazione si trovano alle linee 210-230.
Il problema del "ribaltamento" è stato allora risolto, con l'unico inconveniente di averci fatto perdere il settore originario (lo potremmo tuttavia recuperare rovesciando l'array una seconda volta).

Un'altra possibile soluzione dello stesso problema avrebbe potuto invece essere quella di creare un secondo array (di dimensioni identiche al primo), nel quale copiare gli elementi in ordine inverso. Vediamone direttamente il listato, in quanto lo schema a blocchi è praticamente identico a quello appena visto:

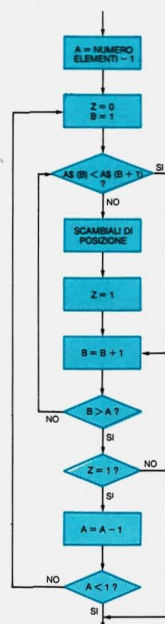
```
10 REM INVERSIONE DI UN VETTORE II
20 INPUT "QUAL È LA DIMENSIONE DEL VETTORE"; N
30 INPUT "L'USCITA LA VUOI SU VIDEO O STAMPANTE (V/S)"; R$
40 R$ = LEFT$ (R$, 1) : REM PRENDE SOLO IL PRIMO CARATTERE
   DI R$
50 IF R$ < > "V" AND R$ < > "S" AND R$ < > "v" AND R$ < > "s"
   THEN GOTO 30
60 IF R$ = "S" OR R$ = "s" THEN OPEN 1, 4 : REM CANALE
   ALLA STAMPANTE
70 IF R$ = "V" OR R$ = "v" THEN OPEN 1, 3 : REM CANALE
   AL VIDEO
80 DIM A (N), B (N) : REM A (N) È IL VETTORE DI PARTENZA
   E B (N) È IL VETTORE INVERTITO
90 REM CARICA IL VETTORE A (N)
100 FOR I = 0 TO N
110 A (I) = INT (RND (0) * N) : REM I NUMERI DELL'ARRAY SARANNO
   COMPRESI TRA 0 ED N
120 NEXT I
130 REM VISUALIZZA IL VETTORE A (N)
140 PRINT # 1, "ELEMENTI ORIGINARI"; CHR$ (13);
150 FOR I = 0 TO N
160 PRINT # 1, A (I); "–";
170 NEXT I
180 REM ------------------------------------
190 REM RIEMPIE L'ARRAY B (N)
200 FOR I = 0 TO N
210 B (N – I) = A (I)
220 NEXT I
225 REM L'ARRAY B (N) È STATO RIEMPITO
230 REM ------------------------------------
240 REM STAMPA L'ARRAY B (N)
250 PRINT # 1, CHR$ (13); CHR$ (13); "ELEMENTI INVERTITI"; CHR$ (13)
260 FOR I = 0 TO N
270 PRINT # 1, B (I); "–";
280 NEXT I
290 CLOSE 1
300 END
```

PROGRAMMAZIONE

L'utilizzo di due vettori (quello di partenza e quello rovesciato) richiede una quantità di memoria esattamente doppia di quella occupata con il primo programma, limitando così le dimensioni massime raggiungibili dagli array stessi. Ciò comporta l'insorgere di un tipico problema: la valutazione della soluzione migliore. Nella stragrande maggioranza delle volte accade infatti che il prezzo da pagare per un qualunque miglioramento nelle prestazioni offerte dai programmi si paghi con un rallentamento nell'esecuzione o con una maggiore occupazione della memoria. Sta allora al programmatore decidere e valutare i pro e i contro relativi a ciascuna soluzione.

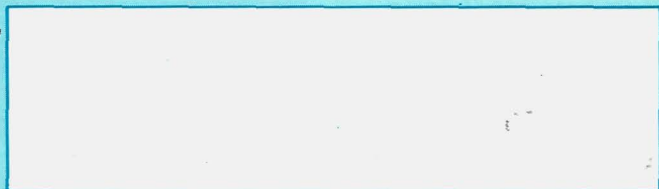
```
10 INPUT "♥QUANTI NOMI"; N : DIMA$ (N)
15 FOR C = 1 TO N : PRINT "NOME"; C
20 INPUT A$ (C) : NEXT C
25 FOR A = N - 1 TO 1 STEP - 1
30 LET Z = 0 : FOR B = 1 TO A
35 IF A$ (B) < A$ (B + 1) THEN GOTO 50
40 LET B$ = A$ (B) : LET A$ (B) = A$ (B + 1)
45 LET A$ (B + 1) = B$ : LET Z = 1
50 NEXT B
55 IF Z = 1 THEN NEXT A
60 PRINT "♥1 VIDEO" SPC (15) "2 STAMPANTE"
65 INPUT "QUALE"; R
70 IF R = 2 THEN OPEN 2, 4 : CMD 2
75 FOR C = 1 TO N : PRINT A$ (C) : NEXT C
80 IF R = 2 THEN PRINT # 2
85 INPUT "ANCORA (S/N)"; R$
90 IF R$ = "S" THEN RUN
```

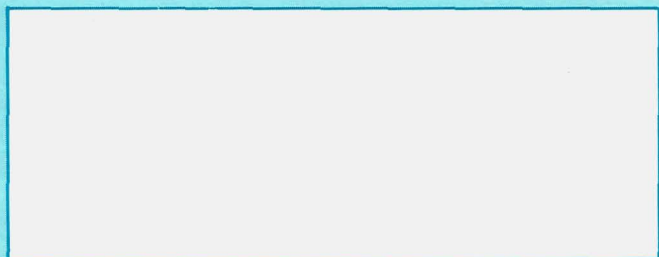
Ordinamento alfabetico

Spesso ci troviamo di fronte a degli elenchi, a delle liste di nomi di persone o di cose. Poterle ordinare alfabeticamente è certamente molto utile per facilitare la ricerca di uno degli elementi. Questo è appunto lo scopo del programma, basato sull'algoritmo noto col nome di "bubble sort".
Si confrontano due elementi contigui: se il primo è minore del secondo, si prosegue; altrimenti si effettua lo scambio di posizione. In tal modo, gli elementi minori vengono via via spinti verso l'alto (all'inizio) della lista, analogamente a quanto accade alle bollicine nell'acqua.
Nota inoltre la possibilità di inviare le informazioni elaborate sullo schermo o su stampante.

Il cuore del programma consiste nella routine di ordinamento (bubble sort) rappresentata nel flow chart.

VIDEOESERCIZI

Quale testo verrà stampato sul video?

```
10 OPEN 9, 3
20 PRINT # 9, "PROVA"
30 PRINT "DI STAMPA"
40 PRINT # 9, "SU XXXXXXXXX"
50 PRINT "DI UN TESTO"
60 PRINT # 9, "QUALSIASI"
70 PRINT "FINE STAMPA"
80 CLOSE 9
90 END
```

Individua quali parole del testo verranno stampate in doppia larghezza. (Se non hai la stampante collegata apparirà alla linea 20 il messaggio di errore "DEVICE NOT PRESENT").

```
10 OPEN 5, 4
20 CMD 5
30 PRINT "QUALE SARA'"
40 PRINT CHR$ (14) "LA"
50 PRINT "PAROLA"
60 PRINT CHR$ (15) "IN"
70 PRINT "DOPPIA
   LARGHEZZA ?"
80 PRINT # 5
90 CLOSE 5 : END
```

Come verrà stampata la stringa che introduci?

```
 10 INPUT "1 – SCHERMO 2 – STAMPANTE"; A
 20 OPEN 5, A + 2
 30 INPUT "INSERISCI UNA STRINGA"; A$
 40 FOR I = 1 TO LEN (A$)
 50 FOR S = 1 TO I
 60 PRINT # 5, " ";
 70 NEXT
 80 PRINT # 5, MID$ (A$, I, 1)
 90 NEXT
100 CLOSE 5
110 END
```